



SQL-Powered Pizza Sales Insights

A Data-Driven Approach Using SQL



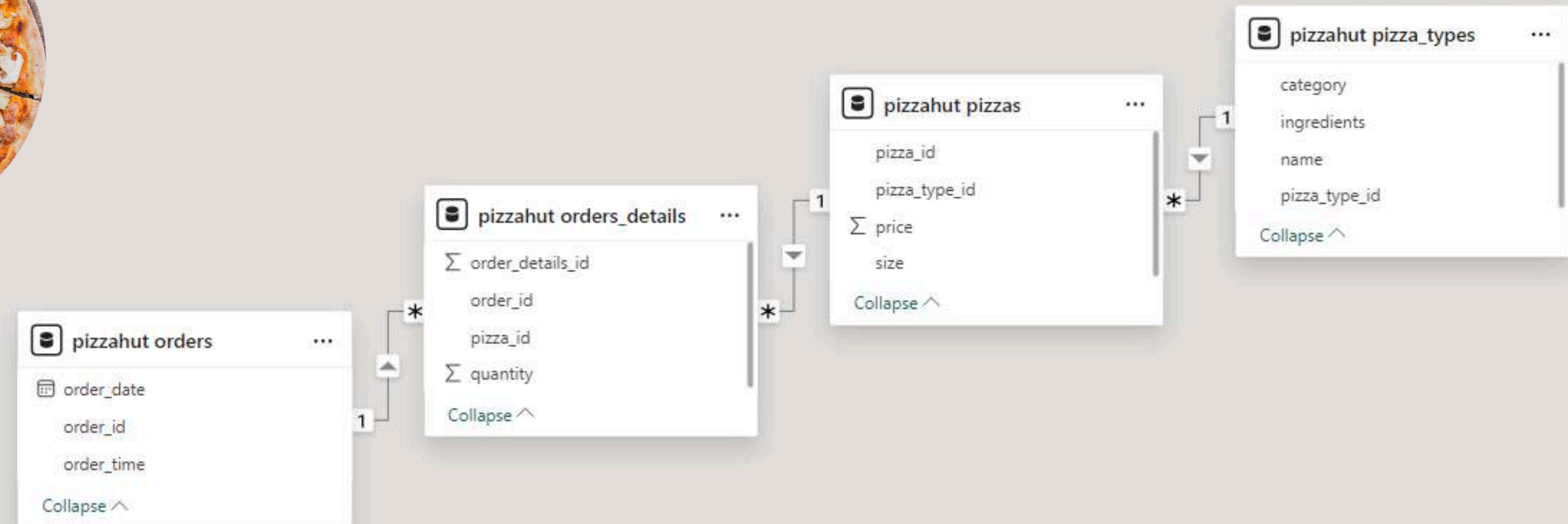
Presented by Saumya Srivastava



PROJECT OVERVIEW

- The objective of SQL-Powered Pizza Sales Insights is to analyze pizza sales data to uncover trends and customer preferences using SQL.
- The project uses a simulated dataset mimicking real-world scenarios.
- Focus areas include total orders, revenue distribution, top-selling items, and advanced sales insights.

DATA MODEL OVERVIEW



KEY TABLES AND COLUMNS



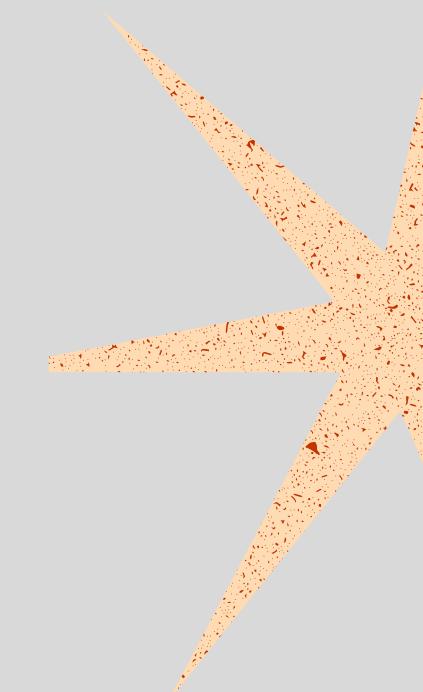
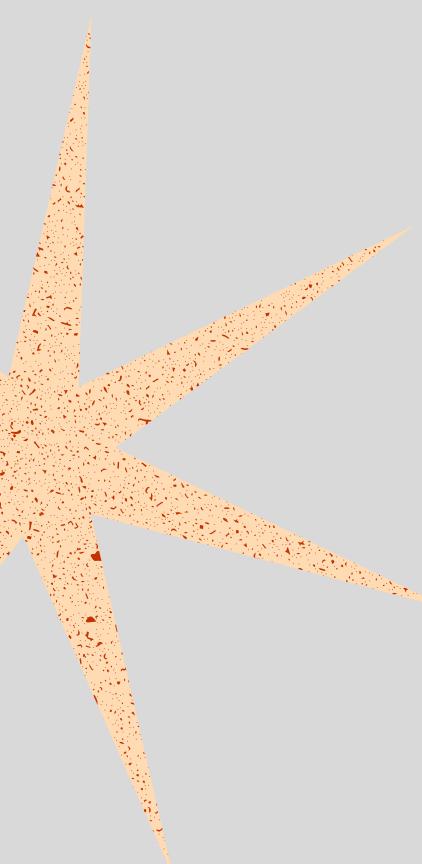
orders_details: order_details_id, order_id, pizza_id, quantity

orders: order_id, order_date, order_time

pizzas: pizza_type_id, pizza_id, size, price

pizza_types: pizza_type_id, name, category, ingredients

THE TOTAL NUMBER OF ORDERS PLACED.



```
Limit to 1000 rows
--- Retrieve the total numbers of orders placed.
• select count(order_id) as total_orders from orders;
```

Result Grid	
	total_orders
▶	21350



THE TOTAL REVENUE GENERATED FROM PIZZA SALES.



```
1 -- the total revenue generated from pizza sales.  
2  
3 • SELECT  
4     ROUND(SUM(orders_details.quantity * pizzas.price),  
5             2) AS total_sales  
6 FROM  
7     orders_details  
8     JOIN  
9         pizzas ON pizzas.pizza_id = orders_details.pizza_id;
```

Result Grid	
	total_sales
▶	817860.05



THE HIGHEST-PRICED PIZZA.

```
1  -- the highest-priced pizza.  
2 • SELECT  
3      pizza_types.name, pizzas.price  
4  FROM  
5      pizza_types  
6      JOIN  
7      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
8  ORDER BY pizzas.price DESC  
9  LIMIT 1;
```

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95



THE MOST COMMON PIZZA SIZE ORDERED.

```
1  -- | the most common pizza size ordered.  
2 • SELECT  
3      pizzas.size,  
4      COUNT(orders_details.order_details_id) AS order_count  
5  FROM  
6      pizzas  
7      JOIN  
8      orders_details ON pizzas.pizza_id = orders_details.pizza_id  
9  GROUP BY pizzas.size  
10 ORDER BY order_count DESC  
11 LIMIT 1;
```

Result Grid		
	size	order_count
▶	L	18526

THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```
1  -- List the top 5 most ordered pizza types along with their quantities.
2  • SELECT
3      pizza_types.name, SUM(orders_details.quantity) AS quantity
4  FROM
5      pizza_types
6      JOIN
7          pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
8      JOIN
9          orders_details ON orders_details.pizza_id = pizzas.pizza_id
10     GROUP BY pizza_types.name
11     ORDER BY quantity DESC
12     LIMIT 5;
```

Result Grid		Filter Rows:
	name	quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY.



```
-- Join the necessary tables to find the total quantity of each pizza category category
SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS total_quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY quantity DESC;
```

	category	total_quantity
▶	Classic	14888
	Veggie	11649
	Supreme	11987
	Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.



```
⚡ ⚡ 🔎 ⏲ ⏳ ⏴ ⏵ | ⏷ | Limit to 1000 rows | ⭐ | 🍗 🔎
-- Determine the distribution of orders by hour of the day.

SELECT
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

Result Grid | Filter Rows:

	hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642
	21	1198
	22	663
	23	28
	10	8
	9	1

JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.



```
1 -- Join relevant tables to find the category-wise distribution of pizzas.  
2 • SELECT  
3     pizza_types.category,  
4     COUNT(pizza_types.name) AS distribution_of_pizzas  
5 FROM  
6     pizza_types  
7 GROUP BY pizza_types.category;
```

Result Grid		Filter Rows:
	category	distribution_of_pizzas
▶	Chicken	6
▶	Classic	8
▶	Supreme	9
▶	Veggie	9

GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.



```
1  -- Group the orders by date and calculate the average number of pizzas ordered per day.
2 • SELECT
3     ROUND(AVG(pizza_ordered), 0)
4   FROM
5     (SELECT
6       orders.order_date,
7         SUM(orders_details.quantity) AS pizza_ordered
8     FROM
9       orders
10    JOIN orders_details ON orders.order_id = orders_details.order_id
11    GROUP BY orders.order_date) AS order_quantity;
```

ROUND(AVG
138

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.



```
1  -- Determine the top 3 most ordered pizza types based on revenue.
2 • select pizza_types.name,sum(orders_details.quantity*pizzas.price) as revenue
3  from pizza_types join pizzas
4  on pizza_types.pizza_type_id=pizzas.pizza_type_id
5  join orders_details
6  on orders_details.pizza_id=pizzas.pizza_id
7  group by pizza_types.name
8  order by revenue desc
9  limit 3
10 ;
11
```

Result Grid | Filter Rows: _____

	name	revenue
▶	The Thai Chicken Pizza	43434.25
▶	The Barbecue Chicken Pizza	42768
▶	The California Chicken Pizza	41409.5

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.



```
1  -- Calculate the percentage contribution of each pizza type to total revenue.
2 • select pizza_types.category,
3   round((sum(orders_details.quantity * pizzas.price) / (select round(sum(orders_details.quantity*pizzas.price),2)
4     from orders_details join pizzas
5       on orders_details.pizza_id=pizzas.pizza_id))*100 ,2)as percentage_contribution
6   from pizza_types join pizzas
7     on pizza_types.pizza_type_id=pizzas.pizza_type_id
8   join orders_details
9     on orders_details.pizza_id=pizzas.pizza_id
10  group by pizza_types.category
11  order by percentage_contribution;
```

Result Grid | Filter Rows:

	category	percentage_contribution
▶	Veggie	23.68
	Chicken	23.96
	Supreme	25.46
	Classic	26.91

THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```
1  -- Calculate the percentage contribution of each pizza type to total revenue.
2 • select pizza_types.category,
3   round((sum(orders_details.quantity * pizzas.price) / (select round(sum(orders_details.quantity*pizzas.price),2)
4     from orders_details join pizzas
5       on orders_details.pizza_id=pizzas.pizza_id))*100 ,2)as percentage_contribution
6   from pizza_types join pizzas
7     on pizza_types.pizza_type_id=pizzas.pizza_type_id
8   join orders_details
9     on orders_details.pizza_id=pizzas.pizza_id
10  group by pizza_types.category
11  order by percentage_contribution;
```

Result Grid | Filter Rows:

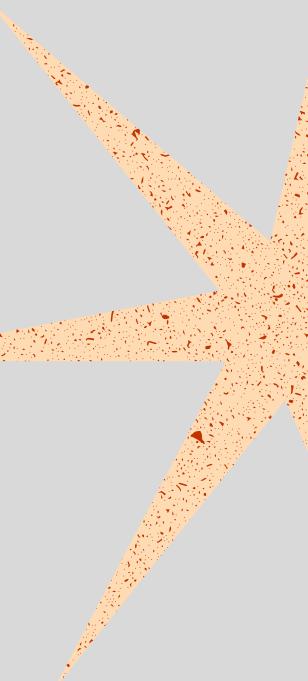
	category	percentage_contribution
▶	Veggie	23.68
	Chicken	23.96
	Supreme	25.46
	Classic	26.91



ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME..



```
1  -- Analyze the cumulative revenue generated over time.  
2 • select sales.order_date,sum(revenue) over(order by order_date) as cum_revenue  
3  from  
4  (select orders.order_date,  
5    sum(orders_details.quantity*pizzas.price) as revenue  
6    from orders join orders_details  
7    on orders.order_id=orders_details.order_id  
8    join pizzas  
9    on pizzas.pizza_id=orders_details.pizza_id  
10   group by orders.order_date) as sales;
```



THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.



```
1  -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
2 • select category, name, revenue
3   from
4     (select category, name, revenue,
5      rank() over(partition by category order by revenue desc) as rn
5   from
6     (select pizza_types.category, pizza_types.name,
7       sum((orders_details.quantity*pizzas.price)) as revenue
8   from pizza_types join pizzas
9   on pizza_types.pizza_type_id=pizzas.pizza_type_id
10  join orders_details
11  on orders_details.pizza_id=pizzas.pizza_id
12  group by pizza_types.category, pizza_types.name) as a) as b
13 where rn<=3;
```



	category	name	revenue
▶	Chicken	The Thai Chicken Pizza	43434.25
	Chicken	The Barbecue Chicken Pizza	42768
	Chicken	The California Chicken Pizza	41409.5
	Classic	The Classic Deluxe Pizza	38180.5
	Classic	The Hawaiian Pizza	32273.25
	Classic	The Pepperoni Pizza	30161.75
	Supreme	The Spicy Italian Pizza	34831.25
	Supreme	The Italian Supreme Pizza	33476.75
	Supreme	The Sicilian Pizza	30940.5
	Veggie	The Four Cheese Pizza	32265.70000000065
	Veggie	The Mexicana Pizza	26780.75
	Veggie	The Five Cheese Pizza	26066.5

CONCLUSION



Comprehensive SQL-Based Analysis

This project utilized SQL to analyze pizza sales data and uncover critical business insights.

Key Insights Gained

Identified total orders, revenue trends, and top-selling pizzas.

Analyzed category-wise and time-based ordering patterns.

Determined top-performing pizzas based on revenue and quantity.

Actionable Outcomes

SQL queries enabled precise data-driven recommendations, such as optimizing inventory, targeting peak hours, and promoting high-revenue items.

Future Scope

Expand the analysis by incorporating additional datasets (e.g., customer feedback or marketing data).

Use advanced SQL techniques for predictive analytics and forecasting.



IT'S PIZZA
TIME!



THANK YOU!

